SMTP/POP3/IMAP Email Engine

Library for Visual FoxPro

Programmer's Manual

(SEE4FP)

Version 7.3

December 9, 2014

This software is provided as-is.
There are no warranties, expressed or implied.

Copyright (C) 2014
All rights reserved

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

MARSHALLSOFT is a registered trademark of MarshallSoft Computing.

TABLE OF CONTENTS

1 Introduction Page 3
  1.1 Features Page 4
  1.2 Documentation Set Page 6
  1.3 Example Program Page 7
  1.4 Installation Page 8
  1.5 Uninstalling Page 8
  1.6 Pricing Page 8
  1.7 Updates Page 8
2 Library Overview Page 9
  2.1 Dynamic Link Libraries Page 9
  2.2 Keycode Page 9
  2.3 Win32/Win64 STDCALL and DECLSPEC Page 9
  2.4 Error Display Page 10
  2.5 INCLUDE Files Page 10
  2.6 FoxPro Forms Page 10
  2.7 Adding SEE to a VFP Program Page 10
  2.8 Dynamic Strings Page 10
  2.9 Null Terminated Strings Page 11
  2.10 Using Internal Memory Page 11
3 Compiler Issues Page 12
  3.1 Compiling Programs Page 12
  3.2 16-bit FoxPro Page 12
  3.3 Compiling to an Executable Page 12
4 Example Programs Page 13
  4.1 Connectionless Example Programs Page 13
  4.2 SMTP Email Example Programs Page 14
  4.3 POP3/IMAP Email Example Programs Page 16
  4.4 IMAP-Only Email Example Programs Page 18
5 Revision History Page 19

1 Introduction

The **SMTP/POP3/IMAP Email Engine for Visual FoxPro (SEE4FP)** library is a toolkit that allows software developers to quickly develop SMTP and POP3/IMAP email applications in Visual FoxPro.

The **SMTP/POP3/IMAP Email Engine (SEE)** is a component DLL library of functions that uses the Windows API to provide direct and simple control of the SMTP (Simple Mail Transport Protocol), POP3 (Post Office 3), and IMAP 4 (Internet Message Access Protocol) protocols.

A straightforward interface allows sending, receiving and parsing email, including multiple MIME base64 and quoted-printable encoded attachments, over any TCP/IP network (such as the Internet). Knowledge of Winsock and TCP/IP is not needed.

The **SMTP/POP3/IMAP Programmer's Manual for Visual FoxPro** provides information needed to compile and run programs in a Visual FoxPro programming environment.

The **SMTP/POP3/IMAP Email Engine for Visual FoxPro** component library supports and has been tested with all versions of Visual FoxPro. **SEE4FP** includes numerous example programs that demonstrate SMTP and POP3/IMAP email functions used to create software applications using the **SEE4FP** library

SEE4FP runs under all versions of Windows (Windows 95, Windows 98, Windows ME, Windows 2000, Windows 2003-2012, Windows NT, Windows XP, Vista, Windows 7 and Windows 8). A Win32 DLL is provided (a 64-bit DLL is available). The **SMTP/POP3/IMAP Email Engine SDK** DLLs (SEE32.DLL or SEE64.DLL) can also be used from any language (C/C++, .NET, Visual Basic, VB. NET, VBA, Delphi, Xbase++, dBASE, COBOL, PowerBASIC, Xbase++, etc.) capable of calling the Windows API.

When comparing the **SMTP/POP3/IMAP Email** component library against our competition, note that:

- SEE4FP is a standard Windows DLL (NOT an OCX or ActiveX control) and is much smaller than a comparable OCX or ActiveX control.
- SEE4FP does NOT depend on ActiveX or similar "support" libraries.
- The WIN32 version of SEE is fully threadable.
- SEE4FP functions can be called from applications not capable of using controls.

MarshallSoft also has versions of the **SMTP/POP3/IMAP Email Engine** library for C/C++ and .NET (SEE4C), Delphi (SEE4D), PowerBASIC (SEE4PB), dBASE (SEE4DB), Xbase++ (SEE4XB), Visual Basic and VB.NET (SEE4VB) and COBOL (SEE4CB). All versions of the **SEE** library use the same Dlls (SEE32.DLL and SEE64.DLL). However, the examples provided for each version are written for the specified programming environment.

The latest versions of **SMTP/POP3/IMAP Email Engine (SEE)** can be downloaded from our web site at

http://www.marshallsoft.com/email-component-library.htm

Our goal is to provide a robust SMTP/POP3/IMAP email component library that you and your customers can depend upon. A fully functional evaluation version is available. Contact us if you have any questions.

1.1 Features

Some of the many features of the **SMTP/POP3/IMAP Email Engine** component library are as follows:

- SMTP client for sending email.
- POP3/IMAP client for receiving email.
- Send email with optional MIME or Quoted Printable attachments.
- Send email with inline embedded HTML, GIF, TIF, JPG, BMP and Rich Text attachments.
- Get the number of messages on the POP3/IMAP email server.
- Get the header lines from any email on the POP3/IMAP email server, without reading the entire email.
- Delete any email on the POP3/IMAP server without reading it first.
- Copy any email on the POP3/IMAP server without deleting it.
- Check for the number of emails on the POP3/IMAP server.
- Easily use with GMAIL/Yahoo/Hotmail servers requiring SSL/TLS.
- Receive any email on the POP3/IMAP server including MIME attachments.
- Forward email.
- Decode email from a file
- Run up to 32 independent WIN32 threads concurrently.
- Can send email to mail addresses on a distribution list.
- Supports SMTP (ESMTP) and POP3 authentication.
- Set return receipt; add TO, CC, BCC recipients
- Set minimum and maximum wait times for server response.
- Supports ISO-8859 (European character sets) and UTF-8 (16 bit character sets) messages.
- Can specify custom Content-Types; add custom header fields
- Includes over 60 functions for SMTP and POP3/IMAP mail control.
- Dozens of switches provided to control how email is sent or received.
- Supports setting priority via X-Priority header field.
- Removes contents of attachments before writing to disk.
- Start and terminate external programs from within an application.
- Is fully thread safe.
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Supports all versions of Visual FoxPro.
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any program (in any language) capable of calling Windows API functions such as C++, Visual C++ NET, dBASE, Delphi, Xbase++, Visual Basic, COBOL, Access and Excel.
- Is compatible with “Managed Code”.
- Works with Windows 95/98/Me/NT/2000/2003/2008/2012/XP/Vista/Win7/Win8
- License covers all programming languages.
- Royalty free distribution (no run-time fees) with your compiled application
- Evaluation versions are fully functional. No unlock code is required.

Registration includes one year of free technical support and updates.

A good selection of Visual FoxPro example programs with full source code is included. Refer to Section 6 for more details on each of the example programs.

[PROGRAM]	[DESCRIPTION]
SEEVER	Displays SEE Version/Build number and registration string.
AUTHEN	Uses authentication to connect to SMTP server.
AUTO	Auto-respond to email using 2 concurrent channels.
BCAST	Sends bulk email to one recipient per email.
CODETEST	Base64 encodes/decodes strings.
FORWARD	Forwards undecoded email.
FROM	Displays header information for email on server. Uses a VFP form.
GB2312	Sends email that is GB2312 (simplified Chinese).
GETRAW	Downloads specified email without decoding.
HELLO	Emails a short message.
HTML	Sends html encoded email with attachments.
ISO8859	Sends ISO-8859 encoded message and subject line
MAILER	Sends email with optional attachment.
MAILER2	Similar to Mailer but uses the “direct” method.
MailSSL	Connects to server requiring SSL to send email.
Mparts	Sends multipart MIME email.
POP3RD	Specifies email message file to read and decode.
READER	Downloads email & attachments and saves to disk.
ReadSSL	Downloads email from POP3 server requiring SSL.
STAT	Lists the number of email messages waiting.
STATUS	Lists all email on server and displays DATE, FROM and SUBJECT Header fields.
STATUS2	Lists all email on server. Similar to STATUS but uses the seeDriver direct method.
ImapFlagsT	Tests manipulation of flaps on IMAP server.
ImapMBTest	Tests IMAP functions.
ImapSEARCH	Tests IMAP search capability.

Also see EXAMPLES.TXT in the DOCS directory for a list of the examples provided for a particular compiler.

1.2 Documentation Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (SEE_4FP) in the set.

- SEE4FP Programmer's Manual (SEE_4FP.PDF)
- SEE User's Manual (SEE_USR.PDF)
- SEE Reference Manual (SEE_REF.PDF)

The SEE_4FP Programmer's Manual (SEE_4FP) is the language specific (Visual FoxPro) manual. All language dependent programming issues such as compiling, compilers and example programs are discussed in this manual.

The SEE User's Manual (SEE_USR) discusses email processing as well as language independent programming issues. Purchasing and license details are also provided.

The SEE Reference Manual (SEE_REF) contains details on each individual SEE function. The manual also contains a list of SEE error codes.

Online documentation can be accessed on the **SMTP/POP3/IMAP Email Engine for Visual FoxPro** product page at:

http://www.marshallsoft.com/see4fp.htm

1.3 Example Program

The following example segment demonstrates the use of some of the **SMTP/POP3/IMAP Email for Visual FoxPro** component library functions: It is recommended that the INCLUDE statements, KEYCODE.FOX and SEE32CON.FOX, be replaced with their contents.

```
#INCLUDE KEYCODE.FOX
#INCLUDE SEE32CON.FOX

DECLARE INTEGER seeAttach in SEE32.DLL INTEGER NbrChans, INTEGER KeyCode
DECLARE INTEGER seeClose in SEE32.DLL INTEGER Chan
DECLARE INTEGER seeErrorText in SEE32.DLL INTEGER Chan, INTEGER Code, STRING
                @Buffer,INTEGER BufLen
DECLARE INTEGER seeIntegerParam in SEE32.DLL INTEGER Chan, INTEGER Param,
                INTEGER Value
DECLARE INTEGER seeRelease in SEE32.DLL
DECLARE INTEGER seeStringParam in SEE32.DLL INTEGER Chan, INTEGER Param,
                STRING @Value

*** PROGRAMMER: Edit these strings [use host name or IP address for server] ***
SmtpServer = "10.0.0.1"
SmtpFrom = "<mike@10.0.0.1>"
SmtpReply = Chr(0)
SmtpTo   = "<mike@10.0.0.1>"
DiagFile = "HELLO.LOG"
*** END PROGRAMMER ***

? "HELLO"
Code = seeAttach(1, SEE_KEY_CODE)
if Code < 0
  ? "Cannot attach SEE"
  return
endif
Code = seeStringParam(0, SEE_LOG_FILE, @DiagFile)
*** set maximum connect wait to 10 seconds
Code = seeIntegerParam(0, SEE_CONNECT_WAIT, 10000)
*** connect to POP3 server
? "Connecting to " + SmtpServer
Code = seeSmtpConnect(0, @SmtpServer, @SmtpFrom, @SmtpReply)
if Code < 0
  Temp = SPACE(128)
  Code = seeErrorText(0,Code,@Temp,128)
  ? Left(Temp,Code)
else
  *** send email message
  ? "Sending email to " + SmtpTo
  Code = seeSendEmail(0,SmtpTo,"","","This is the subject","Message","")
  if Code < 0
    Temp = SPACE(128)
    Code = seeErrorText(0,Code,@Temp,128)
    ? Left(Temp,Code)
  else
    ? "Email has been sent."
  endif
endif
? "Done."
Code = seeClose(0)
Code = seeRelease()
return
```

In the example program above, **seeAttach** is called to initialize **SEE** and then **seeSmtpConnect** is called to connect to the SMTP mail host. **seeSendEmail** is then called, passing the addressee lists. The primary addressee is provided in the "To List". Lastly, the filename of any ASCII or binary attachment is specified. All fields, except the first, in **seeSendEmail** are optional.

After returning from **seeSendEmail**, the **seeClose** function is called to close the connection to the SMTP server. Lastly, **seeRelease** is called to perform **SEE** termination processing and release the Winsock.

1.4 Installation

(1) Before installation of SEE4FP, a Visual FoxPro compiler (any version) should already be installed on your system and tested.

(2) Unzip SEE4FP73.ZIP (evaluation version) or SEExxxxx.ZIP (purchased version where xxxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program SETUP.EXE that will install all SEE4FP files and copy the SEE32.DLL to your Windows directory.

(4) You're ready to compile and run! For a quick start, load project file SEEVER.PRG

1.5 Uninstalling

Uninstalling SEE4FP is very easy.

First, run UINSTALL.BAT, which will delete SEE32.DLL from your Windows directory, typically C:\WINDOWS for Windows 95/98/XP/2003/Vista/Win7/Win8 or C:\WINNT for Windows NT/2000.

Second, delete the **SEE** project directory created when installing SEE4FP.

1.6 Pricing

A developer license for the SMTP/POP3/IMAP Email Library can be registered for $115 USD. Purchasing details can be found in Section 1.4, "How to Purchase", of the SEE User's Manual (SEE_USR).

Also see INVOICE.TXT or

http://www.marshallsoft.com/order.htm

Registration includes one year of free updates and technical support. Registered DLLs never expire.

1.7 Updates

When a developer license is purchased, the developer will be sent a registered DLL plus a license file (SEExxxxx.LIC, where xxxxx is the Customer ID). The license file can be used to update the registered DLL for a period of one year from purchase. Updates can be downloaded from

http://www.marshallsoft.com/update.htm

After one year, the developer license must be updated to be able to download updates and receive technical support. The license can be updated for:

- $30 if the update is ordered within one year of the original purchase (or previous update).
- $55 is the update is ordered between 1 and 3 years of the original purchase (or previous update).
- $75 if the update is ordered after three years of the original purchase (or previous update).

The update price includes **technical support** for an additional year. Note that the registered DLLs, (SEE32. DLL and SEE64.DLL) never expire. Refer to the file UPDATES.TXT located in the /SEE4FP/DOCS directory for more information.

2 Library Overview

The **SMTP/POP3/IMAP Email** component library has been tested on multiple computers running Windows 95/98/Me/XP/2003/2012/Vista/Win7/Win8 and Windows NT/2000.

The SEE4FP library has been tested and works with all versions of 32-bit Visual FoxPro.

The SETUP installation program will copy the SEE DLL to the Windows directory and copies the SEE4FP files to the directory specified (default \SEE4FP). Three sub-directories are created, as follows:

```
DOCS – All documentation files
APPS – All example code
DLLS – All DLL's
```

2.1 Dynamic Link Libraries

The **SMTP/POP3/IMAP Email** component library includes a Win32 dynamic link library (DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLL's have over other "popular" library formats such as VBX or OCX is that DLL's are callable by all Windows applications. Since DLL's are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

2.2 Keycode

SEE32.DLL has a keycode encoded within it. Your keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.FOX. The keycode for the evaluation version is 0. You will receive a new keycode and a new SEE32.DLL after purchasing or updating a developer license. The KEYCODE is passed to **SeeAttach**.

If you get an error message (value -74) when calling **SeeAttach**, it means that the keycode in your application does not match the keycode in the DLL. After registering, it is best to remove the evaluation version of the SEE32.DLL from the Windows search path or delete it.

2.3 Win32/Win64 STDCALL and DECLSPEC

SEE is written in ANSI C and is compiled using the _stdcall and _declspec keywords. This means that SEE4FP uses the same calling conventions and file naming conventions as the Win32/Win64 API. In particular, function names are NOT decorated. Neither leading underscores nor trailing "@size" strings are added to the names of functions.

Any Windows application program may call the SEE32 library provided that the proper declaration file is used.

2.4 Error Display

The error message text associated with **SEE** error codes can be displayed by calling **SeeErrorText**. Each sample program contains examples of error processing.

Also see the file seeErrors.txt for a list of all Winsock and **SEE** error codes.

2.5 INCLUDE Files

All example programs contain two INCLUDE files; `KEYCODE.FOX` and `SEE32CON.FOX`. The file `SEE32CON.FOX` contains all the necessary constants for SEE4FP, while the file `KEYCODE.FOX` contains your keycode, as discussed in Section 2.2.

Since function declarations cannot be in an INCLUDE file (at least through VFP version 5.0), they are listed in each program following the two INCLUDE files (`KEYCODE.FOX` and `SEE32CON.FOX`). The complete list of function declarations is also in the file `SEE32FUN.FOX`

Due to the behavior of Visual FoxPro regarding `INCLUDE` files, it is strongly recommended that the `INCLUDE` files `KEYCODE.FOX` and `SEE32CON.`FOX be replaced with their contents in application programs (i.e., copy and paste contents) of the INCLUDE file.

2.6 FoxPro Forms

SEE functions can be called from any Visual FoxPro code module, such as programs, classes, and forms. See the FROM.SCT example form.

2.7 Adding SEE to a VFP Program

1 - Add the SEE constants (found in SEE32CON.FOX) that will be used in the developer's application.

2 - Add the SEE function declarations (found in SEE32FUN.FOX) that will be called from the developer's application.

Refer to the example programs**.**

2.8 Dynamic Strings

The Visual FoxPro language uses a technique known as "garbage collection" to manage string space at runtime, and may be called internally at any time by the FoxPro runtime, asynchronous to what you may be doing in your code.

When passing a string buffer to a DLL function into which text will be copied, it is strongly recommended that the local string be allocated immediately before use. For example:

```
Code = seeSmtpConnect(0,@SmtpServer,@SmtpFrom,@SmtpFrom)
if Code < 0
  * allocate buffer just before call
  Temp = SPACE(128)
  * put text in Temp
  Code = seeErrorText(1,Code,@Temp,128)
  ? Left(Temp,Code)
endif
```

This technique is not necessary for passing a string to a DLL function, only when passing a buffer to a DLL into which data is to be placed by the DLL function.

2.9 Null Terminated Strings

All strings returned from SEE functions are null terminated. That is, the end of the string is delimited by a Chr(0) character. These strings may be converted for FoxPro in one of two ways: (1) if the length of the string is known, use the FoxPro `LEFT` function: For example,

```
* BASE64 encode
CodedBuff = SPACE(CODED_SIZE)
CodedLen = seeEncodeBuffer(@ClearBuffOne, @CodedBuff, ClearLenOne)
? "CodedBuff:  ", Left(CodedBuff, CodedLen)
```

If the length of the null terminated string is not known, use the FoxPro `AT` function to find the position of Chr(0). For example,

```
N = seeExtractText(@Buffer, @FromText, @Temp, 255)

* extract return address
Pos = AT(":", Temp)
if Pos = 0
  ? "Missing ':' in 'From' or 'ReplyTo' header"
  Code = Close()
  return
endif
```

2.10 Using Internal Memory

This section applies **ONLY** to using **DIRECT** mode as discussed in Section 6 "Theory of Operation" in the SMTP/POP3/IMAP Email User's Manual. (http:/www.marshallsoft.com/see_usr.pdf).

The Visual FoxPro dynamic string management functions (as discussed in Section 2.8 above) have another side effect when running in DIRECT mode (calling **seeDriver**). If Visual FoxPro moves memory at runtime, then memory references by FoxPro will use the new (moved) memory location, although SEE itself will still be using the original memory location previously passed to it. To work around this problem with Visual FoxPro (and other languages that do dynamic string management), you can instruct **seeGetEmailLines** to use its own memory:

```
Code = seeGetEmailLines(Chan, MessageNumber, 0, 0, max_buf_size)
```

If the 4th argument is 0, SEE will use its own memory. After **seeDriver** has been called to completion, the internal buffer can be copied by calling

```
Buffer = SPACE(max_buf_size)
Code = seeDebug(0, SEE_COPY_BUFFER, @Buffer, max_buf_size)
```

seeGetEmailLines is the only function, which requires this technique, since there is no reason to use direct mode in other functions (such as **seeErrorText**) that use return buffers. Refer to the program STATUS2.PRG for an example of using **seeGetEmailLines** in direct mode.

3 Compiler Issues

The **SMTP/POP3/IMAP Email Engine for Visual FoxPro** component library works with all versions of 32-bit Visual FoxPro.

3.1 Compiling Programs

The example programs are compiled from the Visual FoxPro development environment. Before compiling any of the example programs, edit them with your email parameters. Server names can be IP addresses (in decimal dot notation) or the host name. Email addresses must be enclosed in angle brackets.

For more information on host names and email address formats, refer to the SEE User's Manual (SEE_USR). Refer to Section 4.0 below, "Example Programs", for more details on each of the example programs.

3.2 16-bit FoxPro

SEE4FP no longer supports 16-bit FoxPro.

3.3 Compiling to an Executable

FoxPro programs end in ".PRG". They can be compiled to an executable using the FoxPro BUILD command.

For example, to create SEEVER.EXE from SEEVER.PRG in the C:\TEMP directory, type the following in the FoxPro command window:

```
BUILD PROJECT C:\TEMP\SEEVER FROM C:\TEMP\SEEVER
BUILD EXE C:\TEMP\SEEVER FROM C:\TEMP\SEEVER
```

FoxPro executables require VFP500.DLL and VFP5ENU.DLL (ENglish User), and may have to be copied from the VFP CDROM. If you are using an earlier or later version of VFP than version 5.0, substitute the appropriate DLL's for the above.

The FoxPro output display window will disappear as soon as your executable completes. In order to allow the user to control when the display window disappears, add the following code to your application, just before the final return.

```
?  " Type any key to exit..."
X = InKey(0)
```

4 Example Programs

All example programs are written for 32-bit FoxPro. Each has been tested and shows how to correctly use SEE functions. It suggested that the developer compile and run the example programs before developing an application using SEE4FP.

Most of the example programs, with the exception of SEEVER, CODETEST, and FROM, must be edited with your TCP/IP email parameters before compiling. Refer to the SMTP/POP3/IMAP Email User's Manual (online at SEE_USR or /see4fp /DOCS/see_usr.pdf) for details regarding TCP/IP email parameters.

It is highly recommended that INCLUDE statements in the example programs be replaced by their contents before compiling.

Before writing your own programs, compile and run the example programs.

4.1 Connectionless Example Programs

Several example programs do not require a connection to a server.

4.1.1 SEEVER

This simple program displays the SEE version number, build number, and registration string taken from SEE32.DLL. The SEEVER program does not connect to your LAN (or the Internet). Its purpose is to display the SEE version, build, and registration string as well as to verify that SEE32.DLL is being found and loaded by Windows.

This is the first program that you should compile and run.

4.1.2 CODETEST

The CODETEST example program demonstrates how to use **seeEncodeBuffer** and **seeDecodeBuffer**, which BASE64 encodes and decodes several test strings. The CODETEST example program also demonstrates the use of **seeEncodeUTF8** and **seeDecodeUTF8**.

4.2 SMTP Email Example Programs

There are twelve (12) SMTP example programs. SMTP programs send email using an SMTP server. Don't forget to edit the TCP/IP parameters in each program before compiling.

4.2.1 AUTHEN

AUTHEN is an example program that connects to an SMTP server using SMTP Authentication. You must connect to a SMTP server that allows authentication.

AUTHEN.PRG must be edited with your email parameters before compiling.

4.2.2 AUTO

AUTO ("auto-responder") uses two channels concurrently to automatically respond to all new email. AUTO will read (but not delete) all email on your server and reply to each that "your email was received".

AUTO.PRG must be edited with your email parameters before compiling.

4.2.3 BCAST

The BCAST example program emails the same message (BCAST.TXT) to a list of addresses taken from the file, BCAST.EML, containing one email address per line. Along with your SMTP server and your email address, you must create the file containing the email message to send, and create another file containing the list of recipients. See BCAST.EML for an example.

4.2.4 FORWARD

The FORWARD example program forwards an email message to a new recipient. Only undecoded email messages can be forwarded.

4.2.5 GB2312

The GB2312 example program sends a text message that is GB2312 (simplified Chinese) encoded. The recipient's email client will be able to display the email message using the specified GB2312 character set provided that it is capable of identifying GB2312 MIME parts (such as MS OutLook).

4.2.6 HELLO

The HELLO program emails a short message. HELLO.PRG must be edited with your email parameters before compiling.

Compare HELLO with the MAILER example program.

4.2.7 HTML

The HTML example program connects to an SMTP server and emails an HTML file (TEST.HTM) containing inline graphics (IMAGE1.GIF and IMAGE2.GIF). The graphics files are attached to the HTML email message.
HTML.PRG must be edited with your email parameters before compiling.

4.2.8 ISO8859

The ISO8859 example program sends a text message and subject line that is ISO-8859 encoded. The recipient's email client will be able to display the email message using the specified ISO character set provided that it is capable of identifying ISO-8859 MIME parts (such as MS OutLook).

4.2.9 MAILER

The MAILER example program emails a message, including an optional MIME attachment. MAILER.PRG must be edited with your email parameters before compiling.

4.2.10 MAILER2

The MAILER2 example program operates the same as the MAILER program, except that it uses the "direct" method (refer to the SMTP/POP3/IMAP Email User's Manual (online at SEE_USR or /see4fp /DOCS/see_usr.pdf). The function **seeDriver** is called under program control.

Compare to MAILER.PRG.

4.2.11 MailSSL

The MailSSL example program emails a specified email message connecting to a SMTP server that requires SSL, such as Gmail, Hotmail, and Yahoo. Be sure to read the section "Using Stunnel" in the SMTP/POP3/IMAP Email User's Manual (online at SEE_USR or /see4fp /DOCS/see_usr.pdf). MailSSL.PRG must be edited with your email parameters before compiling.

4.2.12 MPARTS

The MParts example program sends a multipart MIME email in which the programmer specifies the Content-Type headers for each attachment.

The two attachment types specified in this example are a sound file (*.wav) and of PDF file (*.pdf).

4.3 POP3/IMAP Example Programs

There are nine (9) POP3/IMAP email example programs. These examples read email from a POP3 (or IMAP) server. Don't forget to edit your email parameters in each program before compiling.

4.3.1 AUTO

AUTO ("auto-responder") uses two channels concurrently to automatically respond to all new email. AUTO will read (but not delete) all email on your server and reply to each that "your email was received".

AUTO.PRG must be edited with your email parameters before compiling.

4.3.2 FROM

The FROM example program is similar to STATUS, except it uses a VFP form to input TCP/IP information at runtime.

From the VFP menu (File/Open), open FROM.PJX. Then, from the project manager dialog box, choose the "Build" button, then "Build Executable" for "Action". After compiling, you will be able to run FROM.EXE.

4.3.3 GETRAW

GETRAW is an example program that downloads a specified email message without decoding it (in "raw" format). This is used to examine what the email looks like on the server. GETRAW.PRG must be edited with your email parameters before compiling.

4.3.4 POP3RD

The POP3RD example program uses the **seePop3Source** function to specify an (undecoded) email message file to be read and decoded.

4.3.5 READER

READER can read email, including multiple MIME attachments, from your POP3 server, optionally deleting each email after being read. READER can also download email without decoding. READER.PRG must be edited with your email parameters before compiling.

4.3.6 ReadSSL

The ReadSSL example program downloads email messages from a POP3 server that requires SSL, such as Gmail, Hotmail, and Yahoo. Be sure to read the section "Using Stunnel" in the SMTP/POP3/IMAP Email User's Manual (online at SEE_USR or /see4fp /DOCS/see_usr.pdf). ReadSSL.PRG must be edited with your email parameters before compiling.

4.3.7 STAT

The STAT example program reads the number of email messages waiting on your POP3 server. STAT.PRG must be edited with your email parameters before compiling.

4.3.8 STATUS

The STATUS example program reads the number of email messages waiting on your POP3 server, and displays the "DATE:", "FROM:", and "SUBJECT:" header fields from each email, as well as the UID for each email message. STATUS.PRG must be edited with your email parameters before compiling.

4.3.9 STATUS2

The STATUS2 example program operates the same as the STATUS program, except that it uses the "direct" method (refer to SMTP/POP3/IMAP Email User's Manual, SEE_USR). The function **seeDriver** is called under program control. STATUS2.PRG must be edited with your email parameters before compiling.

Compare STAT.PRG, STATUS.PRG, and STATUS2.PRG.

4.4 IMAP-Only Example Programs

There are two IMAP-only example programs. These examples access the IMAP server.

4.4.1 ImapFlagsT

The ImapFlagsT example program tests the manipulation of flags on the IMAP server. It reads, sets, and deletes certain flags for the specified email message on the IMAP server.

IMAP flags are:

```
\Seen      Message has been read
\Answered  Message has been answered
\Flagged   Message is "flagged" for urgent/special attention
\Deleted   Message is "deleted" for removal by later EXPUNGE
\Draft     Message has not completed composition (marked as a draft).
\Recent    Message has arrived since the previous time this mailbox was
           selected. ["\Recent" may be fetched but not stored]
```

4.4.2 ImapSearch

The ImapSearch example program tests IMAP search capability.

See ImapSearch.txt or http://www.marshallsoft.com/ImapSearch.htm. for a complete list of all IMAP search strings.

Example search strings as passed to seeImapSearch():

```
SEEN
SEEN NOT ANSWERED
FLAGGED SINCE 1-Feb-2008 NOT FROM "Smith"
LARGER 10000 NOT SEEN
```

5 Revision History

The SMTP/POP3/IMAP Email Engine DLLs (SEE32.DLL and SEE64.DLL) are written in ANSI C. All programming language versions of SEE (C/C++, .NET, Visual Basic, VB .NET, PowerBASIC, Visual FoxPro, Delphi, dBase, Xbase++, COBOL, and Fortran) use the same identical DLLs.

Version 3.0: June 10, 1999.

- Initial release of FoxPro version.

Version 3.1: August 3, 1999.

- Support ISO-8859-1 (Q or B) encoded attachment filenames.
- Support ISO-8859-1 (Q only) on subject line
- SEE_SAVED_TO_MSG added.
- "+OK" line not written to email message file.
- Added seeExtractLine function.

Version 3.2: February 14, 2000.

- Can decode printed quotable attachments!
- seeGetEmailLines can use internal memory.
- Added SEE_WRITE_TO_LOG to seeStringParam.
- Added SEE_GET_ATTACH_NAMES to seeDebug to get attachment filename list.
- Ability to reset the SEE_SET_HEADER header string to "nothing".
- Improvements in dynamic memory usage.
- Added GETRAW and AUTO examples.
- Added seeCommand function.
- Added internal memory usage support.

Version 3.3: November 13, 2000

- seeGetEmailLines can use internal memory.
- Added SEE_COPY_BUFFER [seeDebug] to copy internal buffer.
- Added SEE_WRITE_TO_LOG [seeStringParam] to allow user to write to LOG file.
- Added SEE_GET_ATTACH_NAMES [seeDebug] to get attachment filename list.
- Ability to reset the SEE_SET_HEADER [seeStringParam] to "nothing".
- Added seeCommand function.
- Allow TIC marks (0x27) in VerifyAddressChars().
- Added SEE_GET_LAST_RECIPIENT to seeDebug.
- Added seconds to date string on outgoing email.
- Attachment name is saved when attachment file is closed.
- Added SEE_PATH_DELIMITER to seeIntegerParam().
- Added seeAbort function.
- VerifyFormat rejects "@domain" and "name@" addresses.
- Added "SEE_SET_FROM" so can change "From:" header at runtime.
- Delimiters (CR/LF) sent with command in one network transmission [seeWriteLine].
- Added QUOTED_USER, SEE_SET_CONTENT_TYPE, and SEE_SET_TRANSFER_ENCODING.
- Added SEE_ATTACH_DELIMITER and ability to specify different attachment filename in email.
- Added SEE_ADD_HEADER to seeStringParam.
- Added SEE_WRITE_BUFFER to seeDebug (see seeGetEmailLines)
- Added SEE_ENABLE_IMAGE to send GIF/TIF/BMP/JPG images inside email.

Version 3.4: August 7, 2001

- Supports "AUTH LOGIN" and "AUTH CRAM-MD5" (SMTP) authentication.
- SmtpResponse accepts response line without message.
- Supports ISO-8859-1 (base-64) encoding on subject line.
- Supports "APOP" authentication (POP3 servers).

Version 3.5: March 29, 2002

- Added support for "AUTH PLAIN".
- Recognize multiple AUTH methods on one line, such as "AUTH PLAIN LOGIN CRAM-MD5".
- Added SEE_FORCE_INLINE -- attachments are inline text rather than base64 encoded.
- Added SEE_SET_ATTACH_CONTENT_TYPE -- user can specify content type for attachments.
- Added SEE_ATTACH_BASE_NUMBER -- attachments named "1.att", "2.att", etc.
- Don't close socket (seeClose) if socket is already closed.
- NBR_CHANS set to 128 for Win32.
- SEE_RAW_MODE reads complete lines rather than buffers.
- Added seeQuoteBuffer() -- used to prepare ISO-8859 headers.
- Will continue with sending DATA (rather than return error) if have at least one recipient.
- Call seeStatistics(Chan, SEE_GET_LAST_RECIPIENT) to get # recipients accepted by server.
- Added SEE_IGNORE_REJECTED to ignore error returned if recipient is rejected.
- Added BCAST and CODETEST example program.

Version 3.6: April 14, 2003

- Added seeSendHTML() function.
- Looks for multipart/related as well as multipart/alternative message parts.
- Added SEE_HTML_CHARSET (CHARSET_US and CHARSET_8859)
- Generic multipart boundary definitions handled (not just alternate, related, ...)
- CR/LFs preserved in multiline "Subjects:" headers.
- Handle case where "MIME-Version: 1.0" statement does not proceed all other MIME statements
- MAX_BUF increased from 2048 to 8192 for WIN32
- Virtual socket # written to log file when created (vsGetSocket) & released (vsCloseSocket).
- Write to email file if "MIME-Version" was not seen.
- vSock released properly in seeClose.
- Terminating ALT boundary not written if HTML file is passed from memory (not a file)
- Alternate text in seeSendHTML can be file (if prefixed with '@')
- Added seeEncodeUTF8 and seeDecodeUTF8 functions.
- Delimiters separating email addresses and pathnames changed to a semicolon.
- Added ISO_8859, WIN_1252, and WIN_1255 character set types.

Version 3.7: February 10, 2005.

- Terminating ALT boundary not written if HTML file is passed from memory (not a file).
- Alternate text in seeSendHTML can be file (if prefixed with '@')
- Added seeEncodeUTF8 and seeDecodeUTF8 functions
- AddrDelimiter and PathDelimiter changed to ';' (semicolon)
- Added QUOTED_WIN_1252 and QUOTED_WIN_1255.
- User headers written even if no subject
- Corrected problem: User Content-Type wasn't being sent if no quoting
- Added SEE_HIDE_HEADERS -- overrides any conflicting flags
- Fixed problem with "Filename=" extraction.
- Replaced OF_READ|OF_SHARE_DENY_WRITE with OF_SHARE_DENY_WRITE in _lopen
- Filename added to SEE_CANNOT_CREATE & SEE_CANNOT_OPEN error messages.
- Multi-line subject headers supported in seeGetEmailFile.
- ReadMsgLine uses Allow8Bits to decide if it should quote or not
- Added SEE_SET_DEFAULT_ZONE
- Increased buffer size for challenge string in authenticated SMTP connections.
- Added WriteToLog(), WriteClientTempToLog(), and WriteToLastLog() to centralize log writing.
- Nulls are replaced by spaces in all incoming data.
- Added support for "=?US-ASCII?B?" encoded filenames
- Fixed problem quoting line starting with '.' and having non-ASCII characters.
- Fixed SMTP problem when attaching large number of files (seeWriteSocket,seeWriteLine,seeWriteString).
- Added IgnoreErrorStatus (default TRUE) that skips socket error check in STATE_CONNECT
- Fixed problem with Content-Type prefix (set by SEE_WRITE_CONTENT_TYPE).
- Scan subjects & filenames for "big5" encoding like iso-8859
- Only one of TO, CC, and BCC must contain a recipient.
- Maximum text line length default increased to 1000.
- Added SEE_REPLACE_WITH_COMMAS to override replacement of delimiters with commas.
- SEE_FILE_PREFIX parameters set base for attachment file prefixes.
- Added seeAttachmentParams function.
- Added ISO8859, GB2312, and MParts example programs.

Version 4.0: July 3, 2006.

- Always an error if "relay", "gateway", or "not local" is in the text of the server's response, regardless of SEE_IGNORE_REJECTED.
- Forwarded header lines written to message/rfc822 (attachment) file.
- Each POP3 message optionally saved to disk in raw (undecoded) format in seeGetEmailFile.
- Added function seeForwardEmail().
- Added function seePop3Source().
- Maximum internal buffer size increased from 8 KB to 16 KB.
- Alternate boundaries w/o enclosing quotes are supported.
- FORWARD and Pop3Read example programs added.
- Added function seeByteToShort
- Added function seeShortToByte

Version 4.0: August 10, 2006.

- Always an error if "relay", "gateway", or "not local" is in the text of the server's response, regardless of SEE_IGNORE_REJECTED.
- Forwarded header lines written to message/rfc822 (attachment) file.
- Each POP3 message optionally saved to disk in raw (undecoded) format in seeGetEmailFile.
- Added function seeForwardEmail().
- Added function seePop3Source().
- Maximum internal buffer size increased from 8 KB to 16 KB.
- Alternate boundaries w/o enclosing quotes are supported.
- FORWARD and Pop3Read example programs added.
- Added function seeByteToShort
- Added function seeShortToByte

Version 5.0: May 19, 2008

- Added seeSetErrorText.c example program
- Added LoadLib.c example program.
- Added IMAP capability. IMAP-only functions are:
  1. seeImapConnect : Connect to IMAP server.
  2. seeImapFlags : Get, set, or delete message flags.
  3. seeImapSearch : Search for messages with specified flags.
  4. seeImapMsgNumber : Gets message numbers from buffer filled by seeImapSearch.
  5. seeImapSelectMB : Selects IMAP mailbox.
  6. seeImapDeleteMB : Delete a mailbox.
  7. seeImapCreateMB : Create a new mailbox.
  8. seeImapRenameMB : Rename mailboxes.
  9. seeImapCopyMBmail : Copy messages from selected mailbox to specified mailbox.
  10. seeImapListMB : List all available mailboxes.
- Added ImapFlagsT, ImapSearch, and ImapMBtest example programs.
- Pass NULL for filename to seePop3Source / seeImapSource to revert back to server processing.

Version 5.1: May 20, 2009

- Fixed code for IMAP_SEARCH_MSG_COUNT in seeImapMsgNumber
- Appended CR/LF to text returned by seeGetEmailUID
- Fixed problem with STATE_POP3_DELETE (call exiting via STATE_POP3_DELETE_OK)
- Consider TAB's as ASCII characters (TABIsASCII=1) [bld 6]
- Added EnableHeaders to enable/disable writing of headers.
- Don't write blank line after headers (in STATE_SMTP_BODY) if EnableHeaders = 0
- Write the # bytes written to mail file in the log file.
- Never write boundaries to the email file.
- Fixed bug: seeGetEmailCount works with all IMAP mailboxes (not just InBox)
- Added seeStartProgram and seeKillProgram to start/terminate external programs.
- Fixed problem with blocking mode so connect timeout works.
- Added seeSmtpTarget that writes SMTP output to a file.
- Fixed problem with seeSendEmail (w/ attachment) after forwarding email.
- Added Win64 DLL to support x64. [Visual C++ and Visual Basic version].

Version 5.2: March 20, 2010

- Added seeSleep function (for languages not having a native Sleep call).
- The HELO command passes the computer name rather than its IP address.
- Bug Fix: All handles closed before memory blocks are freed.
- Bug Fix: Multiline "To:" header preserved in incoming email.
- Bug Fix: seeSmtpTarget now always closes files.
- Bug Fix: seePop3Source now always closes files.
- Bug Fix: Multiple IMAP response lines now handled properly by seeCommand.
- Added UTF8 character set support (CHARSET_UTF8).
- Added check for "MX lookup failure" when reading incoming mail.
- Added check for "Invalid MX record" when reading incoming mail.
- Changed IMAP list command argument default from **~/*** to **"" "*".**
- Added SEE_SET_IMAP_LIST_ARG to seeStringParam (sets IMAP list command argument)
- Added seeReadQuoted function: reads a file and quotes the contents as it writes to a buffer.
- Added "Buffer overflow" error code.
- Added QUOTED_ISO_8859_2 to seeIntegerParam for sending ISO_8859_2 encoded emails.
- Added QUOTED_ISO_8859_7 to seeIntegerParam for sending ISO_8859_7 encoded emails.
- Added SEE_GUT_ATTACHMENTS to seeIntegerParam to remove contents of incoming attachments.

Version 6.0: March 1, 2011

- Better integration to the Stunnel proxy server.
- Added seeSmtpConnectSSL and seePop3ConnectSSL.
- Added seeIsConnected.
- Fixed: Can now have leading period in alternate text.
- Added SEE_SET_LOCAL_IP (seeStringParam) to specify local IP.
- Added CHARSET_WIN_1250.
- Changed (default) MaxResponseWait from 10 secs to 25 secs.
- Added SEE_SET_HELO_STRING.
- Fixed problem with reading POP3 from file.
- Add support for ISO-8859-3 and ISO-8859-4.

Version 7.0: November 8, 2011

- Fixed problem decoding some "ISO-8859" subjects
- Fixed problem with wrong content type when using seePop3Rd
- Fixed problem with seeAttachmentParams
- Added seeImapConnectSSL()
- ParseISO removes iso-8859-15 encoding from incoming Subject, etc.
- "To:" and "CC:" strings decoded (base64 & quoted)
- Decode quoted UTF-8 subject strings
- Replace underscore with blank (RFC2047) in UnQuote
- Added ".png" to image types
- Call seeStringParam(Chan, SEE_SET_HELO_STRING, '*') to use machine name for HELO string
- Call seeStringParam(Chan, SEE_LOG_FILE, "\0") to disable logging
- Recognizes iso-2022-jp
- Added seeSetProxySSL()
- Modified seeSmtpConnectSSL(), seePop3ConnectSSL(), seeImapConnectSSL(). Includes changes so that Stunnel (used for email services requiring SSL) is automatically configured, loaded, and unloaded without any user intervention.
- Use large buffer (64K) for IMAP server response on channel 0

Version 7.1: April 10, 2012

- Can pass full pathname for ProxyEXE and ProxyCert in seeSetProxySSL.
- Buffer sizes for ProxyEXE & ProxyCert (seeSetProxySSL) increased from 64 to 256 chars.
- (NOTE: can an no longer pass a null string for PEM certificate)
- seeRelease() kills all running copies of Stunnel started by SEE.
- Password characters not written to log file (PASS ****) & AUTH transmissions
- Added SEE_SET_CONNECT_ATTEMPTS that sets max connection attempts (1 to 12)
- Fixed problem: ImapConnect not returning error if bad login.
- SEE closes all process handles for all external program started by SEE.
-

Version 7.2: September 10, 2013

- Increased the maximum number of channels from 32 to 64.
- Allow multiple subject lines in incoming email.
- Added SEE_REPLACE_UNDERSCORES to seeIntegerParam() to disable replacement of underscores with spaces (RFC2047).
- Fixed problem with GMAIL IMAP connection.
- Can now decode Win1255 subjects.
- seeAbort now always closes attachment files.
- Fixed zone calculation for "half-zones".
- Added debug info to seeGetEmailCount().
- Added STUNNEL_DISABLE_LOGGING flag to seeSetProxySSL() that disables Stunnel logging.
- Fixed problem with SEE_ADD_HEADER when re-opening connection.
- Allow attachment filename to have a leading space.
- Added seeGetHeader() function with parameters SEE_GET_SUBJECT, SEE_GET_FROM, SEE_GET_REPLT_TO, SEE_GET_TO, and SEE_GET_DATE

Version 7.3: December 9, 2014

- Decodes UTF8 encoded attachment filenames.
- Diagnostics written to log file if missing '<' or '>' delimiters in email addresses.
- Added SEE_ALLOW_PARTIAL to seeIntegerParam which allows PARTIAL commands in IMAP.
- Added SEE_GET_UIDVALIDITY to seeStatistics which returns UID Validity in IMAP.
- Fixed problem with boundary buffer [64-bit only].
- Added seeConfigSSL() function which adds lines to the SSL configuration file.
- Added seeUnquote() function that unquotes quoted buffers.
- Added UTF8 quoting : seeIntegerParam(Chan, SEE_QUOTED_PRINTABLE, QUOTED_UTF8)